|  |  |
| --- | --- |
| **自然语言处理及应用**  **实验报告** | |
|  | |
|  |  |
| **名称** | 最小编辑距离 |
| **姓名** | 黄隆宁 |
| **班级** | 硕1224班 |
| **学号** | 3121155049 |
| Email | hln18773372567@stu.xjtu.edu.cn |
| **日期** | 2021-09-15 |

# 实验目的

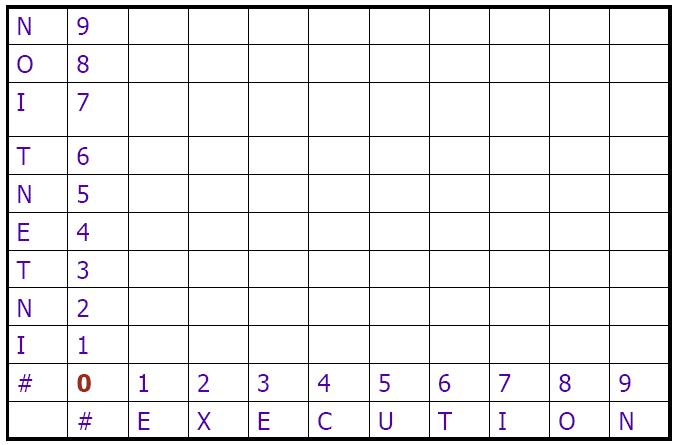
1. 实现英文（及中文）字符串的最小编辑距离，最小编辑路径的计算并显示结果。  
2. 可以输入不同字符串；可以通过修改操作代价来改变最小编辑距离。

# 实验环境

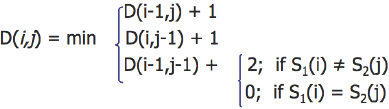
Python3.8

# 实验方法

实验通过二位动态规划算法，实现两字符串之间最小编辑距离计算，在迭代DP矩阵的同时，保存编辑指针，并通过回溯算法求解最小编辑路径，最终打印编辑过程和最小编辑距离的值。



初始化如上图所示的DP矩阵，矩阵的大小m，n分布为两字符串长度+1。

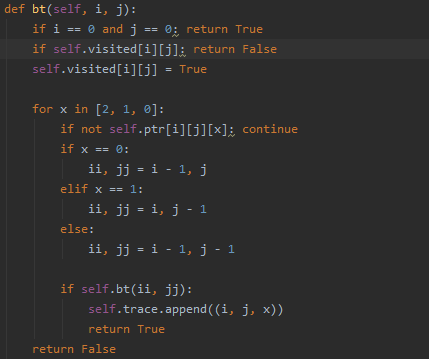


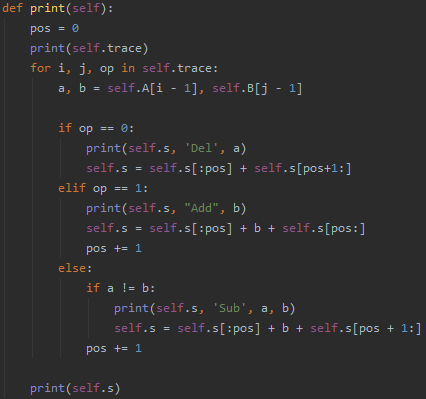
按上图所示规则，逐个迭代DP矩阵，最终得到的D(-1, -1)即为我们所求的最小编辑距离。

为了实现自定义编辑代价，我们只需修改上图中的参数1，1，2。将其修改为我们期望的删除、添加、替换的编辑代价。

同时对于DP矩阵的每一个点，记录所得到的最小值，来自状态转移方程中的哪个或哪些单元格，最为编辑指针。

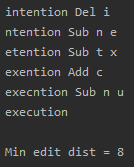
计算编辑路劲部分，根据之前得到的编辑指针，回溯求解得到一条自D(0, 0)到D(-1, -1)的编辑路径。并打印具体的操作过程。



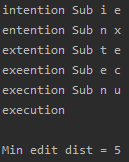


1. **实验结果**

数据'intention', 'execution'，当删除、增加、替换编辑代价分别为1，1，1，2时：



当编辑代价全为1时：



1. **遇到问题及解决思路**

编辑路径打印的编程很麻烦，解决思路就是慢慢写慢慢调。